

In the age of AI, debugging and getting into software engineering has never been easier.

Here are the exact six steps you need to take to unblock yourself from any software engineering dilemma you'll ever encounter, using what we call the AI debugger method:

1. Tinker
2. Ask your AI
3. Read the docs
4. Web search
5. Ask in a forum
6. Ask on a support forum on GitHub
7. Iterate

## AI code debugger - How to debug anything with AI

### 1. Tinker: Pinpoint Your Error

Before you dive into any AI assistance, try to pinpoint your error. An AI buddy (like [Chacha VT](#), [Find](#), or [Bing's AI](#)) could help you here. Pinpointing the parts of your error will potentially solve your problem before you even need an AI. Plus, it allows you to craft a better question when you do turn to AI assistance.

### 2. Ask Your AI: Prompt Engineering Principles

There are six principles to help you get the best out of your AI:

1. Write clear and specific instructions.
2. Give as much context as possible.
3. Use delimiters to indicate distinct parts of the input.
4. Watch out for "hallucinations" (when your AI gives you a wrong output).
5. Understand the limitations of the AI you're working with.
6. Iterate constantly.

AI's are trained on human conversations, so interacting with them should feel like having a conversation. Both Bing's AI and Google's BERT are useful options, but be aware of their limitations. For instance, most AI's have a limit on how many tokens (or words) they keep in context at one time. Practicing asking questions effectively is a skill, and you can check out this free course on [Deep Learning AI](#) to improve your prompt engineering.

### 3. Read the Documentation (with AI Help)

Sometimes documentation can be overwhelming, but you can use AI to your advantage. Copy-paste sections of documentation into your AI's context and ask questions based on it. For example:

The above are the docs for Tool X. Based on those docs, how do I do Y?

By using AI in this way, you'll get more targeted and specific explanations based on your needs.

### 4. Use AI-powered Search Engines

When you face an issue, it's tempting to hope that someone else has run into it before and posted a solution. New AI-powered search engines like [Find](#) combine web search with AI to crawl through site data and provide answers based on what they've read.

## 5. Ask a Forum

If you can't find an existing answer to your question, consider asking a forum like Stack Exchange, Stack Overflow, Reddit, or Git. Make sure to:

1. Format your questions with markdown (you can ask AI for help with formatting).
2. Post your question on a web-indexed forum so that web crawlers and AIs can learn from it. Avoid asking on Discord or Twitter as they don't effectively share knowledge.
3. If the tool you're working on is open-source, use its Git page to ask well-formatted questions. If it's not open-source, reconsider using that tool.

## 6. Ask a support forum

Like GitHub issues, or the groups issue forum.

Or... If you have to, discord.

## 7. Iterate

Repeat these steps as needed. Debugging and problem-solving are iterative processes, so don't be discouraged if you have to go through them several times.

With these six steps, you'll be better equipped to tackle any software engineering challenge that comes your way. Keep hopping through the code and stay riveting, fellow engineers!

## Conclusions

AI is been a complete game changer for developer, making it faster to develop, debug and deploy any piece of code. Following these 7 steps you'll be able to debug probably 99% of code out there.